Simulation-based Real-time Production Control with Different Classes of Residence Time Constraints

Feifan Wang, Feng Ju

Abstract— Residence time constraints are widely observed in production systems, such as semiconductor manufacturing, food industry and battery production, where the time that a part spends in one or several consecutive buffers is restricted. When the residence time of a part is beyond a certain level, the part might face quality problems and need to be further treated or scrapped immediately. To optimize the production performance such as production rate and scrap rate, one needs to properly manage all machines' behavior according to real-time system states to prevent from producing too many intermediate parts with high risk of scrap. To solve this problem, a simulation-based real-time control method is proposed to perform production control in face of four basic classes of residence time constraints that are widely seen in semiconductor manufacturing. A Markov Decision Process (MDP) model is first built, and a feature extraction method and a feature-based approximate architecture are proposed to deal with the curse of dimensionality. Simulation is applied in the training to estimate parameters of the feature-based approximate architecture, so the lookahead function in the MDP model can be approximately obtained. Simulation experiments suggest that such a method leads to significant system performance improvement with low computation overhead, which makes real-time production control feasible for longer serial lines with different classes of residence time constraints.

Index Terms— serial production lines, residence time constraints, simulation-based real-time control, feature-based approximate architecture

I. Introduction

Advances in information and communication technologies enable production systems to respond to production uncertainties quickly and provide potentials to improve the manufacturing efficiency and quality through real-time production control. One production control problem that is commonly seen in real-world factories is to maximize production rate and minimize scrap rate of a production system with residence time constraints [1]. One example of production systems with residence time constraints is semiconductor packaging and testing line, where intermediate semiconductor packages are not allowed to stay in buffer for long primarily for two reasons. First, moisture absorption into polymers of semiconductor packages decreases interfacial adhesion and causes cracks later in reflow process [2]. In addition, long time stay of an intermediate semiconductor package in buffer can lead to oxidation on the surface of its die. Similar problems are also found in semiconductor fabrication [3], [4], [5], [6], food industry [7], [8] and battery production [1].

Feifan Wang was with the School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, 85281 USA. He is now with Robert D. and Patricia E. Kern Center for the Science of Health Care Delivery, Mayo Clinic, Rochester, MN, 55905 USA. Email: Wang.Feifan@mayo.edu

Feng Ju is with School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, 85281, USA. Email: Feng.Ju@asu.edu

The issue of residence time has received mounting attention in production literature in recent years. Paper [9] studies the distribution of residence time of parts in the buffer for a two-machine transfer line, and the risk of scrap is evaluated based on the derived distribution. Such residence time, especially counting from part entry to the system to the departure from the system, is often referred to as lead time or sojourn time in the literature as well. For instance, papers [10], [11], [12] consider lead time in a three-machine transfer line, a production system with closed loop and a two-machine multi-product system, respectively. Paper [13] extends the study on residence time distribution to transfer lines with multiple machines and obtains residence time distribution for each buffer. All these studies on residence time mentioned above assume that defective parts are not scrapped until they finish the last process at the end of the production line. In many applications, parts that violate residence time constraints are scrapped immediately, and it imposes difficulties in capturing system dynamics. Papers [14], [15] take residence time into modeling of two-machine serial lines, and the system dynamics can be captured as defective parts are scrapped immediately. Longer serial lines with defective parts immediately scrapped are studied as an extension of two-machine serial lines. Paper [16] introduces the quality buy rate to model system dynamics and derives steady-state system performance of Bernoulli lines. Papers [17], [18] consider a Bernoulli line where each machine inspects the quality of parts, and parts with residence time larger than a limit have a certain probability of being scrapped. Paper [19] provides a method to evaluate both transient behavior and steady-state behavior of geometric serial lines.

Despite of all the above mentioned efforts, limited work has been reported on real-time production control with residence time constraints due to its complexity. Papers [1], [20] provide methods to perform real-time control, but those methods are only applied to small-sized systems with two machines and one buffer. Thus, one challenge is the large state space of the problem, as one is dealing with a longer serial line. Besides, early studies define residence time constraints for a single buffer or for the whole system, but residence time constraints could have a more complex structure according to what has been observed in semiconductor manufacturing [3]. It leads to another challenge that a proper

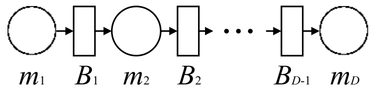


Fig. 1: Serial line

control method is supposed to be flexible to handle different structures of residence time constraints.

In this paper, a simulation-based real-time control method is proposed and intended to overcome the two challenges. Specifically, four basic classes of residence time constraints, covering a wide range of practical applications, are introduced. A Markov Decision Process (MDP) model is formulated, and a feature extraction method and a feature-based approximate architecture are proposed to reduce the state space of the MDP model. Simulation is applied in the training to estimate parameters of the feature-based approximate architecture, so the lookahead function in the MDP model can be approximately obtained. Simulation experiments suggest that such a method leads to significant system performance improvement with low computation overhead, which makes real-time production control feasible for longer serial lines with different classes of residence time constraints.

The rest of the paper is organized as follows. The problem is formulated in Section II, where four basic classes of residence time constraints are introduced. Section III provides the MDP model and a simulation-based method to approximately solve the model. In Section IV, simulation experiments are conducted to validate the proposed simulation-based real-time control method. Finally, this study is concluded in Section V.

II. Problem formulation

A. Serial Production line

The serial line under study is shown in Fig. 1. Parts visit each machine and buffer from the left side to the right side, until they finish all the processes or get scrapped. The following assumptions define the machines, the buffers, and their interactions.

(i) The serial line consists of D machines, denoted by $m_1, m_2, \dots, m_D$, and $(D-1)$ buffers, denoted by $B_1, B_2, \dots, B_{D-1}$.

(ii) All machines are synchronized with a constant processing time (cycle time), which is the time to process a single part.

(iii) Machines are subject to failures. The state of machine m_i, for $i = 1, \dots, D$, is determined at the beginning of a cycle, and it follows the Bernoulli distribution with parameter p_i. Specifically, machine m_i is capable of producing a part in a cycle with probability p_i and fails to do so with probability $(1 - p_i)$. Residence time constraints are usually the concern in practice, when the upstream machines in a serial line have higher efficiency than the downstream machines [20]. Thus, $p_i \geq p_j$ is assumed, for all i and j such that $1 \leq i < j \leq D$.

(iv) Buffer B_i has a finite capacity N_i $(1 \leq N_i < \infty)$, for $i = 1, 2, \dots, D-1$, and its buffer occupancy, denoted by n_i, is determined at the end of a cycle. First-in-first-out (FIFO) policy is assumed regarding the buffer outflow process.

(v) Each part is under residence time constraints, represented by T_{ij}, where $1 \leq i < j \leq D$. Let $\mathcal{T}$ be the set of all residence time constraints. The time that a part spends between the process on machine m_i and the process on machine m_j must be smaller than T_{ij}, if $T_{ij} \in \mathcal{T}$. Otherwise, the part will be scrapped immediately.

(vi) Machine m_i, for $i = 1, 2, \dots, D-1$, is blocked during a cycle, if (a) machine m_i is up, (b) buffer B_i is full, (c) machine m_{i+1} does not produce a part in this cycle due to machine failure or blockage, and (d) there will be no part scrapped from buffer B_i. Machine m_D is never blocked. In addition, block-before-service policy is assumed.

(vii) Machine m_i, for $i = 2, \dots, D$, is starved during a cycle, if machine m_i is up, and buffer B_{i-1} is empty. Machine m_1 is never starved.

(viii) At the end of each cycle, a machine can be turned down manually to prevent it from producing parts in the next cycle. One can also have a machine unchanged, and thus the machine will work as a Bernoulli machine in the next cycle. It is always beneficial to not change the work mode of the last machine. Let $\mathcal{A} \subseteq \{1, 2, \dots, D-1\}$ be the index set of machines that can be turned down. Denote by $a_i(t) \in \{1, 0\}$, for $i \in \mathcal{A}$ and $t = 0, 1, \dots$, the action on machine m_i at the end of cycle t. The action $a_i(t) = 0$ makes machine m_i not work in cycle $(t+1)$. The action $a_i(t) = 1$ represents that machine m_i is unchanged and will work as a Bernoulli machine in cycle $(t+1)$. The control space is denoted by $A = \{0, 1\}^{|\mathcal{A}|}$. Let $\mathbf{a}(t) \in A$ be an action on the serial line in cycle t.

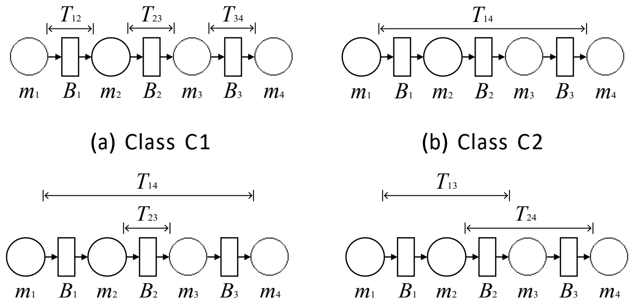
To evaluate and control the serial line, we introduce the performance measures of interest as follows.

- Production rate, $PR(t)$ for $t = 1, 2, \dots$: the expected number of parts produced by machine m_D in cycle t;
- Scrap rate, $SR(t)$ for $t = 1, 2, \dots$: the expected number of parts scrapped from the serial line in cycle t.

In practice, it is desired to have a large production rate $PR(t)$ and a small scrap rate $SR(t)$. The objective of the study is to maximize $(PR(t) - wSR(t))$, where w is a positive constant. This paper studies real-time production control through actions provided by assumption (viii) to improve system performance of a serial line given by assumptions (i) to (vii).

B. Residence time constraints

According to paper [3], we categorize residence time constraints into four basic classes. Fig. 2 shows an example for each class of residence time constraints.



(a) Class C1 (b) Class C2

(c) Class C3 (d) Class C4

Fig. 2: Different classes of residence time constraints

- Class C1. It is a class of residence time constraints between two immediately consecutive processes. Thus, each residence time constraint restricts a buffer, and we have $j = i+1$ for all residence time constraint T_{ij}.
- Class C2. A residence time constraint in this class is between two consecutive but not adjacent processes and puts a limit on a segment of a serial line, consisting of more than one buffer. We have $j > i+1$ for residence time constraint T_{ij}.
- Class C3. In this class, two nested residence time constraints exist. A buffer can belong to two segments restricted by two different residence time constraints. We have $i \;\; k$ and $j \;\; l$ for two different residence time constraints T_{ij} and T_{kl}. This class can be generalized to a case with more than two nested residence time constraints.
- Class C4. An overlapping exists in this class, but two residence time constraints are not nested. Fig. 2d illustrates Class C4 with two different residence time constraints T_{ij} and T_{kl}, where $i < k$, $j < l$ and $k < j$. This class can be generalized to a case with more than two overlapping residence time constraints.

It can be observed that machine m_i, for $i = 1;2; \; ;D$ 1, can be properly controlled (on/off) to improve system performance only when there exists residence time constraint T_{ij} 2 T . Thus, we have $A = f i \; j \, T_{ij}$ 2 T g.

III. Modeling

A. Formulation of MDP model

The state of a part in the serial line can be defined by

$$s = (\mathbf{t}; b); \tag{1}$$

where $\mathbf{t} = [t_1 \quad t_2 \quad\quad]^T$ records the residence time of the part, and b specifies that the part is in buffer B_b.

For a part in a serial line of Class C1 or Class C2, only a single residence time constraint in T has an effect on the part at any time, so only a single residence time is required to be recorded. Thus, $\mathbf{t}$ becomes a one-dimensional vector. For instance, when a new part enters the system, the state of the part is $(t_1 = 0; b = 1)$. The residence time t_1 of the part increases by one each cycle, if the part is restricted by the same residence time constraint. When a part moves out of one residence time constraint and enters a buffer restricted by another residence time constraint, then t_1 is set to be zero. The part is allowed to stay in the system, if the following requirement is satisfied.

$$t_1 \;\; T_{j\,i} \;\; j + b; \text{ for } T_{j\,i} 2 \; T \;\; \text{such that } i \;\; b < \;\; j: \tag{2}$$

Otherwise, the part is scrapped immediately.

For the cases of Class C3 and C4, there are overlapping residence time constraints. Consider a serial line with two overlapping residence time constraints, denoted by T_{ij} and T_{kl}. $\mathbf{t}$ for a part is expressed as $\mathbf{t} = [t_1; t_2]^T$, where t_1 is residence time under constraint T_{ij} and t_2 is residence time under constraint T_{kl}. The initial state of a part is $\mathbf{t} = [0 \quad 0]^T \; ; b = 1$. If the part is in the buffer under constraints T_{ij} and/or T_{kl}, t_1 and/or t_2 will increase by one each cycle. The part is allowed to stay in the system, if the following constraints are satisfied.

$$t_1 \;\; T_{j\,i} \;\; j + b; \text{ if } i \;\; b < \;\; j \tag{3}$$

and

$$t_2 \;\; T_{l\,k} \;\; l + b; \text{ if } k \;\; b < \;\; l: \tag{4}$$

Let $H(t)$ be the set of the states of all parts in the serial line in cycle t. We use $H(t)$ to define the system state of the serial line. Denote by H the state space. In this model, the initial state $H(0)$ is assumed to be known, and then the MDP model is introduced as follows.

- Reward function: The reward function, for $t = 1; 2; ,$ is denoted by $r(H(t \;\; 1); \mathbf{a}(t \;\; 1))$. Specifically,

$$r(H(t \;\; 1); \mathbf{a}(t \;\; 1)) = P\tilde{R}(t) \;\; wS\tilde{R}(t); \tag{5}$$

where $P\tilde{R}(t)$ and $S\tilde{R}(t)$ are the number of parts produced by machine m_D and the number of parts scrapped from the serial line in cycle t, respectively. As the action $\mathbf{a}(t \;\; 1)$ is being taken, both the number of produced parts and the number of scrapped parts in cycle t are unknown. Thus, $P\tilde{R}(t)$ and $S\tilde{R}(t)$ are random variables.
- The expected total discounted reward of policy p for any initial state $H(0)$:

$$v^p(H(0)) = E^p \sum_{i=0}^{¥} l^i r(H(i); \mathbf{a}(i)) \;\; ; \tag{6}$$

where l 2 $[0; 1)$ is the discount, and control policy is a map $p : H \; ! \; A$.
- The optimal expected total discounted reward:

$$v(H(0)) = \max_p E^p \sum_{i=0}^{¥} l^i r(H(i); \mathbf{a}(i)) \;\; : \tag{7}$$

- The optimal control policy:

$$p \; 2 \; \arg\max_p E^p \sum_{i=0}^{¥} l^i r(H(i); \mathbf{a}(i)) \;\; : \tag{8}$$

If the optimal expected total discounted reward $v(H(t))$ is known for any state $H(t)$ 2 H , the optimal action at the end of cycle $(t-1)$ can be obtained as follows.

$$\mathbf{a}(t-1) \in \arg\max_{\mathbf{a}(t-1)\in A} E\Big[r(H(t-1); \mathbf{a}(t-1)) + \lambda v(H(t)) \Big]. \tag{9}$$

The value iteration and the policy iteration are two widely used iterative methods to solve MDP model and estimate $v(H(t))$ and $\mathbf{a}(t)$. However, the model in this paper cannot be solved by either method directly due to its large state space. Alternatively, a simulation-based method will be introduced to address the problem.

B. Feature-based approximate architecture

The consideration of residence time constraints brings a large state space even for a two-machine serial line [20]. In the problem in this paper with multiple machines, the state space is much larger, and it is impossible to obtain $v(H(t))$, making it infeasible to obtain the optimal action through Equation (9). To address the problem, we apply the feature-based architecture to reduce the dimensionality of the problem and introduce an approximate lookahead function $\hat{v}(\phi(H(t)); \boldsymbol{\beta})$ to replace $v(H(t))$. Function $\phi(H(t))$ represents the feature extraction that maps system state $H(t)$ into a feature vector. The lookahead function is then approximated by linearly weighting the features. Specifically,

$$\hat{v}(\phi(H(t)); \boldsymbol{\beta}) = \boldsymbol{\beta}^T \phi(H(t)); \tag{10}$$

where $\boldsymbol{\beta}$ is a vector of parameters estimated through simulation, and the details are presented in Section III-C. Thus, Equation (9) can be rewritten as follows.

$$\mathbf{a}(t-1) \in \arg\max_{\mathbf{a}(t-1)\in A} E\Big[r(H(t-1); \mathbf{a}(t-1)) + \lambda \hat{v}(\phi(H(t)); \boldsymbol{\beta}) \Big]. \tag{11}$$

Due to the large state space, the optimal control π that maps system state $H(t)$ to the optimal action $\mathbf{a}(t)$ cannot be written explicitly as a lookup table. Therefore, $\boldsymbol{\beta}$ is stored to represent the optimal control policy π to map system state to the optimal action through Equation (11).

The number of parts in the segment of the serial line restricted by a residence time constraint and the residence time of the head part in the segment are two important features of the segment [1], [20]. Paper [21] develops a simulation model of Class C1 to study the effect of features on the system performance. The simulation experiment suggests that the buffer occupancy should not be either too small or too large. Small buffer occupancy reduces the production rate because of a high probability of starvation for the downstream machines. Large buffer occupancy increases risk of scrap. In addition, the simulation experiment also shows that a small residence time is always preferred. Thus, three features are adopted to describe the segment of the serial line limited by a residence time constraint. They are the number of parts in the segment, the square of the number of parts in the segment and the residence time of the head part in the segment. With a constant term included, the dimension of both $\phi(H(t))$ and $\boldsymbol{\beta}$ is $(3|T|+1)$, and $\phi(H(t))$ is expressed as follows.

$$\phi(H(t)) = \begin{bmatrix} f_1 & f_2 & \cdots & f_{3|T|+1} \end{bmatrix}^T. \tag{12}$$

The first feature f_1 is set to be one as a constant term. Let the ith residence time constraint in T be T_{jk}, for $i = 1, 2, \dots, |T|$. The features of the segment of the serial line covered by T_{jk} are

$$\begin{aligned} f_{3i-1} &= \sum_{l=j}^{k-1} n_l; \\ f_{3i} &= \left(\sum_{l=j}^{k-1} n_l \right)^2; \\ f_{3i+1} &= \tau; \end{aligned} \tag{13}$$

where τ is the value of the dimension in $\boldsymbol{\tau}$, corresponding to residence time constraint T_{jk}, of the head part in the segment.

C. Training and implementation

The purpose of the training is to estimate parameter $\boldsymbol{\beta}$ through simulation, so $\hat{v}(\phi(H(t)); \boldsymbol{\beta})$ can become the estimate of $v(H(t))$ and replace $v(H(t))$ in Equation (9). The procedure of the training is provided as follows.

Step 1: Initialize the setting for the training. Set the number of iterations of the policy iteration method to be I. A total number of simulation runs is set to be K in each iteration of the policy iteration method. Simulation starts from cycle 0 and ends in cycle J.

Step 2: Let the index i be one. Set the initial control policy, denoted by π^0, to be a policy that never turns any machine down.

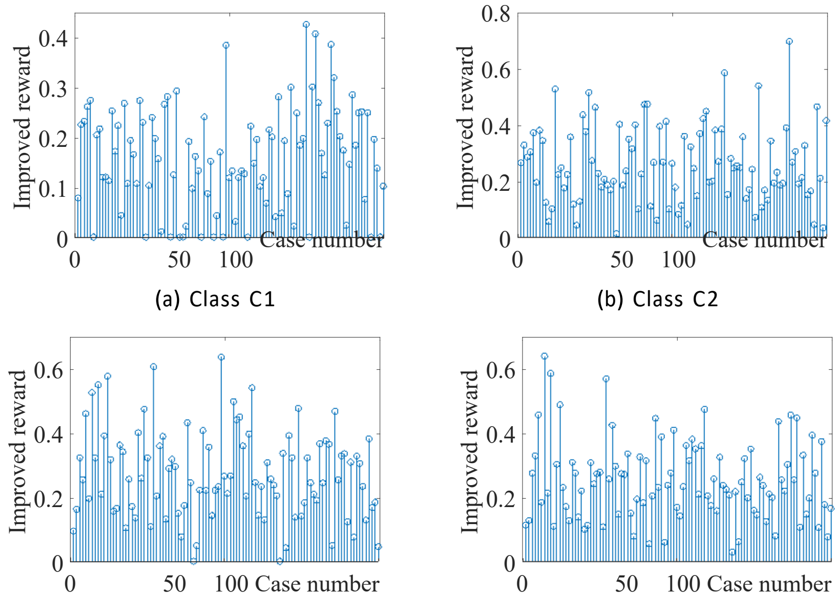
Step 3: Start the ith iteration of the policy iteration method. Randomly generate K initial system states, denoted by $H^k(0)$ for $k = 1, 2, \dots, K$. Control policy π^{i-1} is applied to each simulation run. The parameter for the ith iteration, denoted by $\boldsymbol{\beta}^i$, is estimated through least squares estimation as follows.

$$\boldsymbol{\beta}^i \in \arg\min_{\boldsymbol{\beta}} \sum_{k=1}^{K} \left(\hat{v}\left(\phi\left(H^k(0)\right); \boldsymbol{\beta}\right) - \sum_{j=1}^{J} \lambda^{j-1} r\left(H^k(j-1); \mathbf{a}(j-1)\right) \right)^2; \tag{14}$$

where $\sum_{j=1}^{J} \lambda^{j-1} r\left(H^k(j-1); \mathbf{a}(j-1)\right)$ represents the total discounted reward of the realization of the kth simulation run.

Step 4: The control policy for the ith iteration, π^i, is obtained and represented using parameter $\boldsymbol{\beta}^i$. The maps from system state $H(t-1)$ to the optimal action under control policy π^i is obtained as follows.

$$\mathbf{a}(t-1) \in \arg\max_{\mathbf{a}(t-1)\in A} E\Big[r(H(t-1); \mathbf{a}(t-1)) + \lambda \hat{v}\left(\phi(H(t)); \boldsymbol{\beta}^i\right) \Big]. \tag{15}$$



(a) Class C1 (b) Class C2

(c) Class C3 (d) Class C4

Fig. 3: Improved reward by simulation-based control

Step 5: Increase index i by one. If the index i is greater than I, go to step 6. Otherwise, go to step 3.

Step 6: Validate the control policy. There is a small chance that the training does not go in the improving direction, especially when the size of the problem is large and the number of residence time constraints is large. The obtained control policy will only be applied if it delivers system performance improvement according to the simulation.

The training is performed offline. The resulting control policy π is represented by the estimated $\boldsymbol{\beta}$. To implement the control policy at the end of cycle $(t-1)$ with an observed system state $H(t-1)$, the optimal action is obtained online by Equation (11). The computing time to run Equation (11) is small, and small reaction time of the system can be guaranteed to support real-time capability. Serial lines of different classes need to be trained separately but following the same procedures.

IV. Performance Evaluation

The simulation-based real-time control method is developed with MATLAB, and the experiment runs on a computer with Intel(R) Core(TM) i7-8700 CPU, 16 GB RAM, and 64-bit Windows 10 Enterprise operating system.

To test the proposed method in a more general sense, parameter settings are randomly selected from a given range shown as follows.

$$\begin{aligned} &D \in \{4,5,6\};\\ &p_i \in [0.5, 0.99] \text{ for } i=1,\dots,D;\\ &N_i \in \{4,5,6,7\} \text{ for } i=1,\dots,D-1;\\ &w \in [0.7, 1.7]. \end{aligned} \tag{16}$$

p_i, for $i = 1,\dots,D$, is sorted in a descent order. Discount λ is set to be 0.95. Simulation runs 300 cycles each time from an initial state with all buffer empty. Residence time constraints are randomly generated. For Class C1, we have

$$T_{i\,i+1} \in \{N_i+1, N_i+2, N_i+3\} \text{ for } i=1,\dots,D-1. \tag{17}$$

For Class C2, we have

$$T_{1D} \in \left\{\sum_{k=1}^{D-1} N_k+1, \sum_{k=1}^{D-1} N_k+2, \sum_{k=1}^{D-1} N_k+3\right\}. \tag{18}$$

We define two residence time constraints for serial lines of Class C3 as follows.

$$T_{1D} \in \left\{\sum_{k=1}^{D-1} N_k+1, \sum_{k=1}^{D-1} N_k+2, \sum_{k=1}^{D-1} N_k+3\right\}, \tag{19}$$

and

$$T_{ij} \in \left\{\sum_{k=i}^{j-1} N_k+1, \sum_{k=i}^{j-1} N_k+2, \sum_{k=i}^{j-1} N_k+3\right\}, \tag{20}$$

where $i \in [1, D-1] \cap \mathbb{Z}$ and $j \in [i+1, i+\min(D-i, D-2)] \cap \mathbb{Z}$. In the experiment for serial lines of Class C4, the residence time constraints are selected in the range shown as follows.

$$T_{1i} \in \left\{\sum_{k=1}^{i-1} N_k+1, \sum_{k=1}^{i-1} N_k+2, \sum_{k=1}^{i-1} N_k+3\right\}, \tag{21}$$

and

$$T_{jD} \in \left\{\sum_{k=j}^{D-1} N_k+1, \sum_{k=j}^{D-1} N_k+2, \sum_{k=j}^{D-1} N_k+3\right\}, \tag{22}$$

where $i \in [3, D-1] \cap \mathbb{Z}$ and $j \in [2, i-1] \cap \mathbb{Z}$.

TABLE I: Performance of simulation-based control

Class	Average reward per cycle without control	Average reward per cycle with control	Average relative improvement with control
C1	0.2331	0.3849	65:12%
C2	0.2680	0.5502	105:30%
C3	0.2739	0.5244	91:46%
C4	0.2972	0.5359	80:32%

We randomly generate 200 parameter settings for each class of serial lines. Simulation for each parameter setting runs 300 cycles with 100 replications. The average reward $(PR(t) \quad wSR(t))$ of each cycle from cycle 101 to 300 in each repeat is calculated and compared. Fig. 3 shows that the improved rewards of most cases can be beyond 0.1 for serial lines of Class C1 and greater than 0.15 for the three other classes. TABLE I presents statistics for further comparison. The second column of the table presents the average reward where all machines keep working with no control to turn a machine down. The third column gives the average reward of the proposed method. The average relative improvement with control is calculated and listed on the table, showing a large improvement of the simulation-based real-time control.

V. Conclusions

Residence time of parts is commonly restricted in many production systems. Due to the complexity of these systems, resulting from the large state space of the problem and diverse structures of residence time constraints, it is difficult to perform real-time production control to improve production performance such as production rate and scrap rate. This paper is intended to contribute to this end. Four basic classes of residence time constraints, covering a wide range of practical applications, are introduced. A feature extraction method and a feature-based approximate architecture are proposed to reduce the complexity of the problem and obtain real-time production control policy. Simulation experiments suggest significant system performance improvement of such a method. The future research can be directed to investigating structural properties of each class of residence time constraints, and thus the simulation-based control method can improve further through feature selection and feature-based approximate architecture. In addition, it is worth exploring the performance of the proposed method in mathematical models of production systems with different assumptions, such as queueing network, serial line with a more general machine reliability model, etc., to further determine the impact of different assumptions on feature selection.

References

[1] F. Ju, J. Li, and J. A. Horst, “Transient analysis of serial production lines with perishable products: Bernoulli reliability model,” IEEE Transactions on automatic control, vol. 62, no. 2, pp. 694–707, 2016.

[2] B. Han and D.-S. Kim, “Moisture ingress, behavior, and prediction inside semiconductor packaging: A review,” Journal of Electronic Packaging, vol. 139, no. 1, 2017.

[3] A. Klemmt and L. Mönch, “Scheduling jobs with time constraints between consecutive process steps in semiconductor manufacturing,” in Proceedings of the 2012 Winter Simulation Conference (WSC), pp. 1–10, IEEE, 2012.

[4] F. Yang, N. Wu, Y. Qiao, M. Zhou, and Z. Li, “Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 3, pp. 502–516, 2016.

[5] C. Pan, M. Zhou, Y. Qiao, and N. Wu, “Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges,” IEEE transactions on automation science and engineering, vol. 15, no. 2, pp. 586–601, 2017.

[6] J. Wang, H. Hu, C. Pan, Y. Zhou, and L. Li, “Scheduling dual-arm cluster tools with multiple wafer types and residency time constraints,” IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 3, pp. 776–789, 2020.

[7] P. Amorim, H. Meyr, C. Almeder, and B. Almada-Lobo, “Managing perishability in production-distribution planning: a discussion and review,” Flexible Services and Manufacturing Journal, vol. 25, no. 3, pp. 389–413, 2013.

[8] M. Shin, H. Lee, K. Ryu, Y. Cho, and Y.-J. Son, “A two-phased perishable inventory model for production planning in a food industry,” Computers & Industrial Engineering, vol. 133, pp. 175–185, 2019.

[9] C. Shi and S. B. Gershwin, “Part waiting time distribution in a two-machine line,” IFAC Proceedings Volumes, vol. 45, no. 6, pp. 457–462, 2012.

[10] C. Shi and S. B. Gershwin, “Lead time distribution of three-machine two-buffer lines with unreliable machines and finite buffers,” SMMSO 2015, p. 211, 2015.

[11] A. Angius, M. Colledani, A. Horváth, and S. B. Gershwin, “Analysis of the lead time distribution in closed loop manufacturing systems,” IFAC-PapersOnLine, vol. 49, no. 12, pp. 307–312, 2016.

[12] A. Angius and M. Colledani, “Analysis of the lead time distribution in multi-product systems with dedicated buffers,” IFAC-PapersOnLine, vol. 51, no. 11, pp. 1624–1629, 2018.

[13] C. Shi and S. B. Gershwin, “Part sojourn time distribution in a two-machine line,” European Journal of Operational Research, vol. 248, no. 1, pp. 146–158, 2016.

[14] F. Ju, J. Li, et al., “Transient analysis of bernoulli serial line with perishable products,” IFAC-PapersOnLine, vol. 48, no. 3, pp. 1670–1675, 2015.

[15] N. Kang, F. Ju, and L. Zheng, “Transient analysis of geometric serial lines with perishable intermediate products,” IEEE Robotics and Automation Letters, vol. 2, no. 1, pp. 149–156, 2016.

[16] R. Naebulharam and L. Zhang, “Bernoulli serial lines with deteriorating product quality: performance evaluation and system-theoretic properties,” International Journal of Production Research, vol. 52, no. 5, pp. 1479–1494, 2014.

[17] J.-H. Lee and J. Li, “Performance evaluation of bernoulli serial lines with waiting time constraints,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 1087–1092, 2017.

[18] J.-H. Lee, J. Li, and J. A. Horst, “Serial production lines with waiting time limits: Bernoulli reliability model,” IEEE Transactions on Engineering Management, vol. 65, no. 2, pp. 316–329, 2017.

[19] F. Wang and F. Ju, “Transient and steady-state analysis of multistage production lines with residence time limits,” IEEE Transactions on Automation Science and Engineering, vol. 18, no. 1, pp. 122–134, 2020.

[20] F. Wang, F. Ju, and N. Kang, “Transient analysis and real-time control of geometric serial lines with residence time constraints,” IISE Transactions, vol. 51, no. 7, pp. 709–728, 2019.

[21] F. Wang and F. Ju, “Decomposition-based real-time control of multi-stage transfer lines with residence time constraints,” IISE Transactions, vol. 53, no. 9, pp. 943–959, 2021.